Convolutional Soft Decision Trees

Alper Ahmetoğlu[1] Ozan İrsoy[2] Ethem Alpaydın[1]

[1]Department of Computer Engineering
Boğaziçi University
İstanbul, Turkey

[2]Bloomberg LP
NY, U.S.A.

October, 2018

Soft decision trees

Response of a binary decision tree node m:

$$F_m(\boldsymbol{x}) = F_{ml}(\boldsymbol{x})g_m(\boldsymbol{x}) + F_{mr}(\boldsymbol{x})(1 - g_m(\boldsymbol{x})) \tag{1}$$

In a hard decision tree, $g_m(\boldsymbol{x}) \in \{0, 1\}$.
In a soft decision tree, $g_m(\boldsymbol{x}) \in [0, 1]$, where

$$g_m(\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^T\boldsymbol{x})}} \tag{2}$$

Leaves contain constant values, $\boldsymbol{\rho}_m$. They can be also parameterized by adding a linear projector, $\boldsymbol{\rho}_m = V\boldsymbol{x}$.
Also known as hierarchical mixtures of experts (Jordan and Jacobs, 1993).

Because of this we can fit to data smoothly with fewer number of nodes.

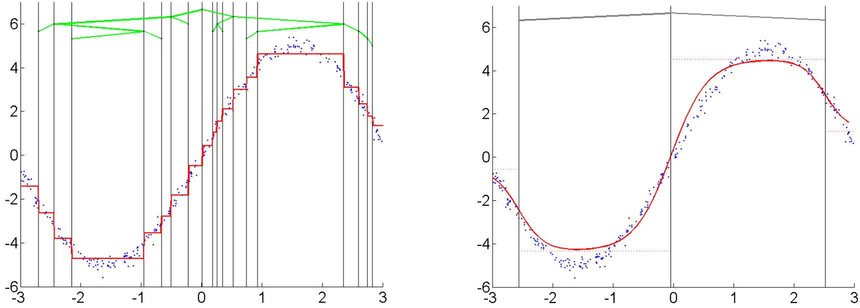
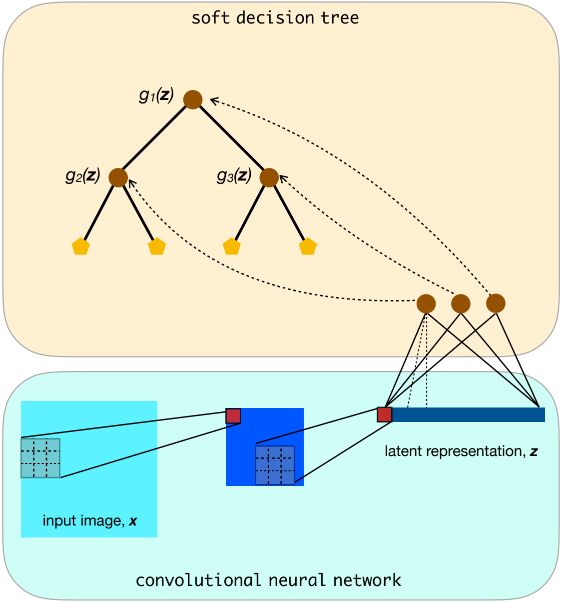


Figure: A hard decision tree (left) and a soft decision tree (right). Reprinted from İrsoy et al. 2012.

Convolutional soft decision trees

- A more complex gating function results in a more complex model, therefore brings representational advantage.
- We can choose any differentiable $g(\boldsymbol{x})$.
- In this work, we choose $g(\boldsymbol{x})$ to be a convolutional neural network.


soft decision tree
$g_1(\boldsymbol{z})$
$g_2(\boldsymbol{z})$
$g_3(\boldsymbol{z})$
latent representation, $\boldsymbol{z}$
input image, $\boldsymbol{x}$
convolutional neural network

Regularization of soft decision trees

- When the representational power of $g(\boldsymbol{x})$ increases model becomes prone to overfitting.
- Previously, L^2 and L^1 regularizations for soft decision trees are examined and L^2 is reported to work slightly better (Yıldız et al. 2013).
- We compare L^2 regularization with input dropout regularization.

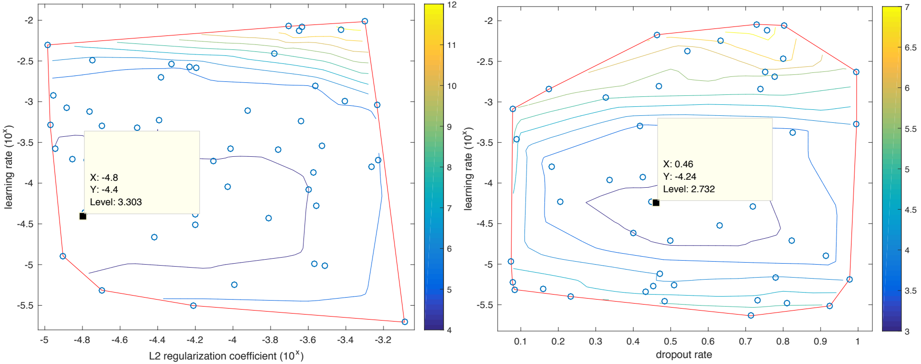


Figure: Error surfaces with respect to different hyperparameter settings.

$dim(z)$	SDT-3	SDT-4	SDT-5	SDT-L3	SDT-L4	SDT-L5	MLP-8	MLP-16	MLP-32
MNIST									
Orig. $\boldsymbol{x}$	11.96	7.99	7.51	2.67	2.57	**2.30**	7.76	4.74	3.16
50	1.37	1.08	0.76	0.72	0.71	0.63	0.56	0.54	**0.52**
100	1.02	0.96	0.98	0.66	0.67	0.74	**0.59**	0.61	0.59
200	1.11	0.84	0.95	0.76	0.76	0.62	0.68	**0.55**	0.57
Fashion-MNIST									
Orig. $\boldsymbol{x}$	20.95	29.80	20.83	11.94	11.50	**11.35**	16.66	14.50	13.47
50	10.46	10.24	10.56	7.36	**7.28**	8.08	8.02	7.55	7.73
100	10.12	10.40	9.76	7.89	**7.36**	8.05	8.16	7.67	7.56
200	12.28	9.14	10.37	7.55	7.18	**7.08**	7.59	7.51	7.81
CIFAR-10									
50	9.38	9.52	9.18	8.85	8.76	**8.64**	8.94	8.66	8.99
100	9.71	9.27	9.67	8.83	8.72	8.96	9.02	**8.69**	9.07
200	11.83	10.90	9.95	8.91	9.60	9.75	9.16	9.01	**8.85**

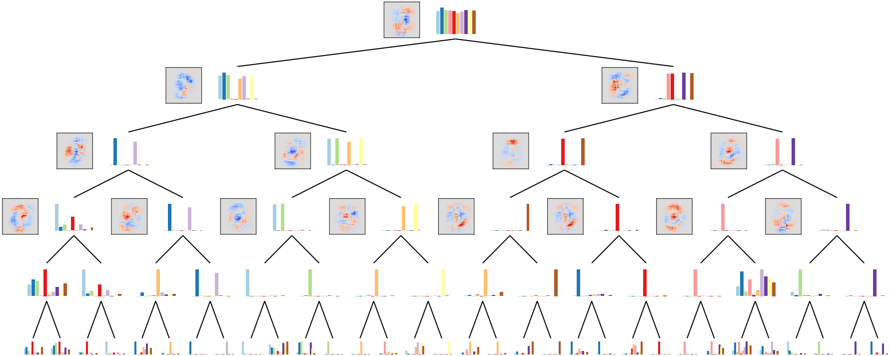


Figure: Colored vertical bars represent class distributions on each decision node for MNIST. On the left of decision nodes are average gradients w.r.t. input (red is high, blue is low).

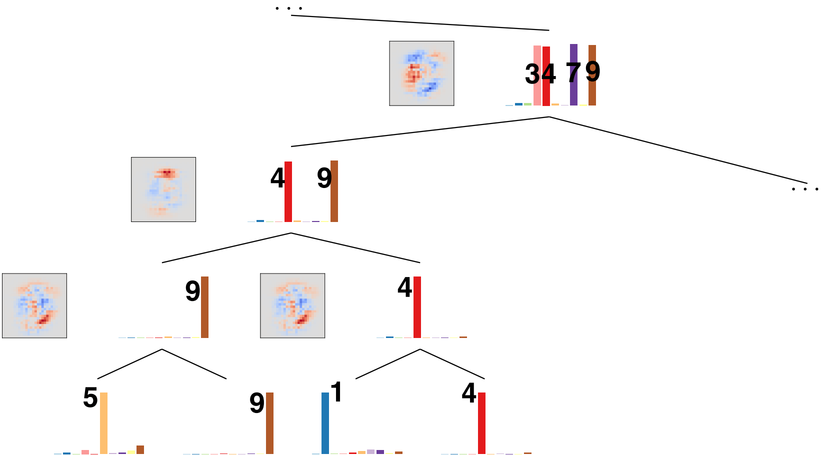


Figure: red: positively high, blue: negatively high, gray: low

Conclusions

- CSDT performs comparable to a CNN with dense layers.
- CSDT is interpretable. We can analyze its hierarchical decisions.
- Dropout regularization in SDTs is slightly better than L^2 regularization.

Thank you for your attention.
Questions are welcome.